Sistema de Monitoramento de Trens Automatizado

Davi Vasconcelos Souza

Leonardo Carvalho Jeronimo Souza

Luis Felipe Diniz Santos

Challenge CCR, Sprint 1

São Paulo

2024

Sumário

Descritivo

O projeto tem como objetivo criar um sistema de monitoramento de trens em tempo real utilizando Machine Learning para a detecção de anomalias e a previsão de falhas. O setor de transportes ferroviários é crucial para o deslocamento de pessoas e mercadorias, e qualquer interrupção nos serviços pode causar grandes prejuízos econômicos e transtornos aos usuários. A justificativa para o desenvolvimento deste sistema está na crescente demanda por soluções que permitam um monitoramento proativo e preventivo, substituindo abordagens reativas tradicionais.

Monitorar trens em tempo real possibilita a identificação de padrões anômalos antes que se tornem falhas críticas, o que é essencial para evitar paradas inesperadas e otimizar as rotinas de manutenção. Um sistema automatizado e inteligente pode reduzir significativamente o tempo de inatividade dos trens e aumentar a eficiência operacional.

A previsão de falhas baseada em dados históricos e atuais garante que as equipes de manutenção possam agir preventivamente, melhorando a segurança e a confiabilidade do sistema.

Modelagem de Classes

- **Train (Trem)**
    - Atributos:
        - trainId: String
        - status: Status
        - lastMaintenanceDate: LocalDate
    - Métodos:
        - startTrain()
        - stopTrain()
- **Sensor**
    - Atributos:
        - sensorId: String
        - type: SensorType
        - location: String
    - Métodos:
        - collectData()
- **AnomalyDetector**
    - Métodos:
        - detectAnomaly(SensorData data): Anomaly
- **Anomaly**
    - Atributos:
        - anomalyId: String
        - description: String
        - severity: SeverityLevel
    - Métodos:
        - getSeverity()
- **Alert**
    - Atributos:
        - alertId: String
        - anomaly: Anomaly
        - timestamp: LocalDateTime
    - Métodos:
        - sendAlert()

- **Operator (Operador)**
    - Atributos:
        - operatorId: String
        - name: String
    - Métodos:
        - acknowledgeAlert(Alert alert)
- **MaintenanceRequest**
    - Atributos:
        - requestId: String
        - train: Train
        - issueDescription: String
    - Métodos:
        - submitRequest()
- **Dashboard**
    - Atributos:
        - dashboardId: String
        - alerts: List<Alert>
    - Métodos:
        - displayData()
- **PerformanceReport**
    - Atributos:
        - reportId: String
        - reportData: List<Train>
    - Métodos:
        - generateReport()
- **MaintenanceSchedule**
    - Atributos:
        - scheduleId: String
        - train: Train
        - scheduledDate: LocalDate
    - Métodos:
        - scheduleMaintenance()

Diagrama UML